

A Low-Complexity Joint Detection-Decoding Algorithm for Nonbinary LDPC-Coded Modulation Systems

Xuepeng Wang, Baoming Bai
State Key Lab. of ISN, Xidian University
Xi'an 710071, China
E-mail: wangxuepengxd@gmail.com
bmbai@mail.xidian.edu.cn

Xiao Ma
Department of ECE, Sun Yat-sen University
Guangzhou, GD 510275, China
E-mail: maxiao@mail.sysu.edu.cn

Abstract—In this paper, we present a low-complexity joint detection-decoding algorithm for nonbinary LDPC coded-modulation systems. The algorithm combines hard-decision decoding using the message-passing strategy with the signal detector in an iterative manner. It requires low computational complexity, offers good system performance and has a fast rate of decoding convergence. Compared to the q -ary sum-product algorithm (QSPA), it provides an attractive candidate for practical applications of q -ary LDPC codes.

I. INTRODUCTION

Nonbinary low-density parity-check (LDPC) codes were first introduced by Gallager in [1] based on modulo arithmetic. In [2], Davey and MacKay presented a class of nonbinary LDPC codes defined over finite field $\text{GF}(q)$ with $q > 2$. They also introduced a sum-product algorithm (SPA) for decoding q -ary LDPC codes, named QSPA. Now, it has been shown that nonbinary LDPC codes have better performance than binary LDPC codes [2], [3], especially when combined with higher-order modulations. Recently, a surge appears in the study of nonbinary LDPC codes [5], [6], [7], [8], [9].

However, the advantages of nonbinary LDPC codes over its binary counterpart are balanced by their higher decoding complexity. To reduce the decoding complexity, Davey and MacKay proposed a more efficient QSPA, called fast Fourier transform based QSPA (FFT-QSPA), for decoding LDPC codes over $\text{GF}(2^p)$ [4]. Moreover, a simplified decoding algorithm called extended min-sum (EMS) was proposed by Declercq and Fossorier in [5] to further reduce decoding complexity. It provides a good candidate for decoding q -ary LDPC codes with small q . For larger field size (say, $q > 32$), the sort operations required by the EMS algorithm will incur higher complexity. As a result, the decoding complexity is still a concern for practical implementation of q -ary LDPC coded systems.

Most recently, Mobini *et al* [13] and Huang *et al* [15] developed reliability-based decoding algorithms for binary LDPC codes with low complexity. Motivated by their work and [14], this paper will explore hard-decision based decoding for q -ary LDPC codes, and present a low-complexity joint

detection-decoding algorithm for q -ary LDPC-coded modulation systems, which provides efficient trade-off between system performance and implementation complexity.

The algorithm is devised to combine the simplicity of hard-decision decoding with the good performance of message-passing algorithms. In the proposed scheme, signal detection and decoding are integrated as a whole, and the input signal vector to detector is updated in an iterative way. At each iteration, the updated hard-decision results from detector are delivered to the LDPC decoder which performs hard-decision decoding using message-passing algorithm. The output of decoder is then fed back to detector, with which the received signal points are updated such that they are progressively close to the transmitted signals in observation space. This can be viewed as an iterative denoising processing. Compared to the FFT-QSPA, the proposed algorithm requires lower computational complexity and has fast rate of decoding convergence.

II. NONBINARY LDPC-CODED MODULATIONS

A. System Model

The nonbinary LDPC-coded modulation system under consideration is shown in Fig. 1. Assume that an LDPC code $\mathcal{C}[N, K]$ over $\text{GF}(q)$ with $q > 2$ is used in conjunction with a two-dimensional signal constellation \mathcal{X} of size $|\mathcal{X}|$. The input vector of information symbols, $\mathbf{u} \in \text{GF}(q)^K$, is first encoded by the LDPC encoder into a codeword $\mathbf{v} = (v_0, v_1, \dots, v_{N-1}) \in \mathcal{C}$. The corresponding code rate $R_c = K/N$. The codeword \mathbf{v} is then mapped to \mathcal{X} , producing the modulated signal vector $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})$ with $x_j = \mathcal{M}(v_j) \in \mathcal{X}$, where $\mathcal{M}(\cdot)$ stands for the signal mapping function. In this paper, we always assume the constellation size is equal to the finite field size, i.e., $|\mathcal{X}| = q$. The spectral efficiency for this coded-modulation system is

$$\rho = R_c \log_2 |\mathcal{X}| \quad \text{bits/signal.} \quad (1)$$

Suppose that the complex signal vector \mathbf{x} is transmitted over the AWGN channel. The received vector $\mathbf{y} = (y_0, y_1, \dots, y_{N-1})$ is then given by

$$y_j = x_j + n_j, \quad j = 0, 1, \dots, N-1, \quad (2)$$

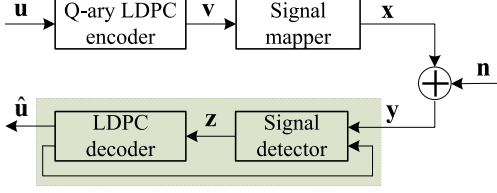


Fig. 1. A nonbinary LDPC-coded modulation system

where $n_j \sim \mathcal{CN}(0, N_0)$ are independent and identically distributed complex Gaussian random variables with zero mean and variance $N_0/2$ per dimension. Denote by $E_s = \mathbf{E}[|x_j|^2]$ the average energy per transmitted signal. Then the average received signal-to-noise ratio (SNR) is

$$\text{SNR} = E_s/N_0 = \rho E_b/N_0,$$

where E_b denotes the average energy per information bit.

In this paper, we consider a hard-decision based iterative detection-decoding strategy. In each iteration the signal detector makes hard-decision about \mathbf{v} based on the updated received vector \mathbf{y} , producing vector $\mathbf{z} = (z_0, z_1, \dots, z_{N-1})$ with $z_j \in \text{GF}(q)$; then the decoder performs hard-decision decoding with \mathbf{z} as the input. The hard extrinsic-information produced by the decoder is then fed back to the signal detector to update \mathbf{y} . We will show that with this decoding strategy, good system performance can be achieved with reduced complexity.

B. LDPC Codes over $\text{GF}(q)$

A q -ary LDPC code \mathcal{C} of length N over $\text{GF}(q)$ is given by the null space of a sparse $M \times N$ parity-check matrix $\mathbf{H} = [h_{i,j}]$ over $\text{GF}(q)$, where M is the number of check equations and $M = N - K$ if \mathbf{H} is full rank. Let $\mathbf{v} = (v_0, v_1, \dots, v_{N-1})$ be a codeword in \mathcal{C} . Then the parity-check constraints can be expressed as $\mathbf{v}\mathbf{H}^T = \mathbf{0}$, or

$$\sum_{j=0}^{N-1} h_{i,j} v_j = 0, \quad i = 0, 1, \dots, M-1, \quad (3)$$

where the operations of multiplication and addition are all defined over $\text{GF}(q)$. If the matrix \mathbf{H} has constant row weight d_c and constant column weight d_v , then the corresponding code is called a (d_v, d_c) -regular q -ary LDPC code.

Similar to its binary counterpart, a q -ary LDPC code can also be described using a Forney-style factor graph [16], as depicted in Fig. 2, where \mathcal{M} denotes the mapper/demapper. The graph has N variable nodes corresponding to coded symbols, and M check nodes corresponding to parity-check equations. For convenience, we define the following two index sets

$$\mathcal{N}_v(j) = \{i | h_{i,j} \neq 0, 0 \leq i < M\}, \quad (4)$$

$$\mathcal{N}_c(i) = \{j | h_{i,j} \neq 0, 0 \leq j < N\}. \quad (5)$$

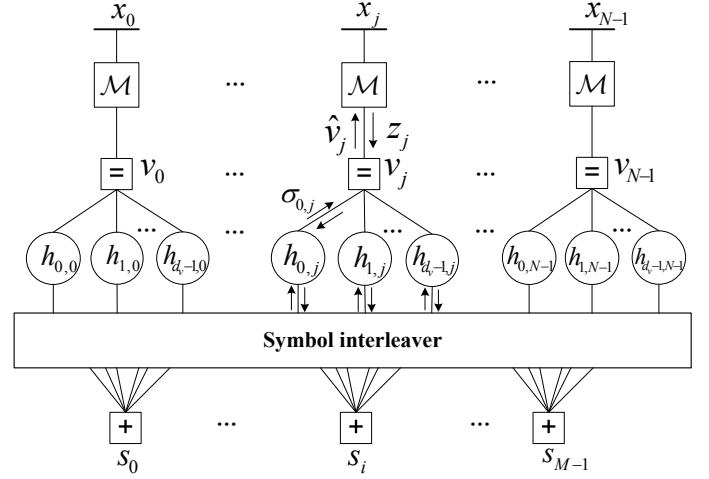


Fig. 2. Forney-style factor graph of the nonbinary LDPC-coded system

III. JOINT DETECTION-DECODING ALGORITHM FOR q -ARY LDPC-CODED SYSTEMS

In this section, we will develop an iterative joint detection-decoding algorithm for the LDPC-coded modulation system shown in Fig. 1. The whole algorithm is a hard-decision based message-passing algorithm (MPA) operating on the factor graph. Assume that a (d_v, d_c) -regular q -ary LDPC code is used.

A. Signal Detection and Message-Passing Based on Hard Information

Let $\hat{\mathbf{x}} = (\hat{x}_0, \hat{x}_1, \dots, \hat{x}_{N-1})$ with $\hat{x}_j \in \mathcal{X}$ denote the estimate of \mathbf{x} made by the signal detector based on the received vector \mathbf{y} . Assume that all the constellation points in \mathcal{X} are used equal likely. Then with the maximum likelihood decision rule, the detected signal $\hat{\mathbf{x}}$ is given by

$$\hat{x}_j = \underset{x \in \mathcal{X}}{\text{argmin}} \|y_j - x\|, \quad j = 0, 1, \dots, N-1, \quad (6)$$

where $\|\cdot\|$ denotes the Euclidean (l_2) norm. The input of the decoder \mathbf{z} is simply the demapping output of $\hat{\mathbf{x}}$, i.e.,

$$z_j = \mathcal{M}^{-1}(\hat{x}_j) \in \text{GF}(q), \quad j = 0, 1, \dots, N-1. \quad (7)$$

The vector \mathbf{z} is a codeword if and only if its M -tuple syndrome $\mathbf{s} = (s_0, s_1, \dots, s_{M-1})$ equals $\mathbf{0}$, i.e.,

$$\mathbf{s} \equiv \mathbf{z}\mathbf{H}^T = \mathbf{0}. \quad (8)$$

For $0 \leq i < M$, the component s_i of \mathbf{s} is given by

$$s_i = \sum_{j \in \mathcal{N}_c(i)} h_{i,j} z_j, \quad (9)$$

which is called a *check-sum* of received symbols. A received symbol z_j is said to be checked by s_i if $h_{i,j} \in \text{GF}(q) \setminus \{0\}$. From (9) with $s_i = 0$, the estimate of v_j given by other variable nodes participating in the check-sum s_i can be expressed as

$$\sigma_{i,j} = -h_{i,j}^{-1} \left(\sum_{j' \in \mathcal{N}_c(i) \setminus j} h_{i,j'} z_{j'} \right). \quad (10)$$

This is the *update rule for check-to-variable node message*. Since the column weight of \mathbf{H} is d_v , every symbol v_j can receive d_v estimates along the set of edges $\{(i, j) | i \in \mathcal{N}_v(j)\}$, as depicted in Fig. 2. As can be seen from (9) and (10), only hard information propagates between variable and check nodes, i.e., the variable nodes send their hard-decision decoded symbols to the check nodes, and the check nodes simply compute the syndromes and send estimates back to their adjacent variable nodes. The estimate $\sigma_{i,j}$ for v_j can be considered as the extrinsic information [15].

B. Iterative Detection-Decoding

We now proceed to consider the update rule for variable nodes. Refer to Fig. 2. Assume that a variable node has received the extrinsic information from adjacent check nodes. Different from general message-passing algorithms, we will use this information to update the received samples to improve the reliability measure of received signal. To do this, an iterative process is performed between variable nodes and signal detector. For simplicity, the proposed iterative joint detection-decoding algorithm will be referred to as IJDD hereafter.

In the following, we first introduce some notations used for the IJDD algorithm. Let k_{max} be the maximum number of iterations to be performed. For $0 \leq k < k_{max}$, let:

- $\mathbf{y}^{(k)} = (y_0^{(k)}, y_1^{(k)}, \dots, y_{N-1}^{(k)})$ be the input vector to the signal detector in the k th iteration; $\hat{\mathbf{x}}^{(k)}$ and $\mathbf{z}^{(k)}$ be the corresponding detected signal vector and the output decision vector.
- $\mathbf{s}^{(k)} = (s_0^{(k)}, s_1^{(k)}, \dots, s_{M-1}^{(k)})$ be the syndrome of $\mathbf{z}^{(k)}$.
- $\sigma_{i,j}^{(k)}$ be the extrinsic information for v_j given by the i th check-sum involving v_j in the k th iteration.
- $\mathcal{D}(y_j^{(k)}, r)$ denote a valid search sphere of radius r centered at $y_j^{(k)}$.
- $\vec{\mathbf{L}}_j^{(k)}(\mathbf{p}, \mathbf{q}) = \mathbf{q} - \mathbf{p}$ be a correction vector directed from the point \mathbf{p} to the point \mathbf{q} . For brevity, we will use $\vec{\mathbf{L}}_j^{(k)}$ for $\vec{\mathbf{L}}_j^{(k)}(\mathbf{p}, \mathbf{q})$.
- $f_j^{(k)}(a)$, $a \in \text{GF}(q)$ denote the number of occurrences of the element a in $\{\sigma_{i,j}^{(k)}\}_{i \in \mathcal{N}_v(j)}$.

Clearly, $0 \leq f_j^{(k)}(a) \leq d_v$ and $\sum_{a \in \text{GF}(q)} f_j^{(k)}(a) = d_v$. $f_j^{(k)}(a)$ indicates a reliability measure for decoding z_j into the symbol a . Let

$$a_{max} = \arg \max_{a \in \text{GF}(q)} \{f_j^{(k)}(a)\}, \quad (11)$$

and

$$\Delta f_j^{(k)} = f_j^{(k)}(a_{max}) - \max_{a \in \text{GF}(q) \setminus \{a_{max}\}} \{f_j^{(k)}(a)\},$$

where a_{max} is the element in $\text{GF}(q)$ that has the highest reliability for v_j , and $\Delta f_j^{(k)}$ represents the difference in number of votes between the two highest-voted candidates for v_j in the k th iteration. With the plurality voting rule, we choose

$$\hat{v}_j^{(k)} = a_{max}.$$

With the above discussions, the *message update rule for variable nodes* can be formulated as follows.

Message Update Rule for Variable Nodes: Make estimation on $\hat{v}_j^{(k)}$ using (11) based on $\{\sigma_{i,j}^{(k)}\}_{i \in \mathcal{N}_v(j)}$, $j = 0, 1, \dots, N-1$, and evaluate $f_j^{(k)}(a_{max})$ and $\Delta f_j^{(k)}$. Then the variable nodes send the triples $(\hat{v}_j^{(k)}, \Delta f_j^{(k)}, f_j^{(k)}(a_{max}))$, $0 \leq j \leq N-1$, to detector, where the following operations are performed on $y_j^{(k)}$:

$$y_j^{(k+1)} = y_j^{(k)} + \xi_j^{(k)} \vec{\mathbf{L}}_j^{(k)}, \quad (12)$$

Here $\vec{\mathbf{L}}_j^{(k)}$ and $\xi_j^{(k)}$ are given based on $\hat{x}_j^{(k)}$ and $(\hat{v}_j^{(k)}, \Delta f_j^{(k)}, f_j^{(k)}(a_{max}))$, $0 \leq j \leq N-1$, which can be described as follows.

- If $\mathcal{M}(\hat{v}_j^{(k)}) \in \mathcal{D}(y_j^{(k)}, r)$, then

$$\vec{\mathbf{L}}_j^{(k)} = \begin{cases} \hat{x}_j^{(k)} - y_j^{(k)}, & \text{if } \mathcal{M}(\hat{v}_j^{(k)}) = \hat{x}_j^{(k)} \\ \mathcal{M}(\hat{v}_j^{(k)}) - \hat{x}_j^{(k)}, & \text{if } \mathcal{M}(\hat{v}_j^{(k)}) \neq \hat{x}_j^{(k)} \end{cases} \quad (13)$$

and

$$\xi_j^{(k)} = \begin{cases} f_j^{(k)}(a_{max})/d_v, & \text{if } \Delta f_j^{(k)} \geq T \\ \Delta f_j^{(k)}/d_v, & \text{if } \Delta f_j^{(k)} < T \end{cases} \quad (14)$$

- Otherwise set $\vec{\mathbf{L}}_j^{(k)} = \mathbf{0}$ and $\xi_j^{(k)} = 0$.

Note that $\mathcal{D}(y_j^{(k)}, r)$ specifies a region where x_j is located most likely. Only $\mathcal{M}(\hat{v}_j^{(k)})$ within this region are used to update $y_j^{(k)}$. In this paper we set r to be $1.415d_{min}$ and T to be 3, where d_{min} is the minimum Euclidean distance among constellation points. Moreover, the case of $\mathcal{M}(\hat{v}_j^{(k)}) = \hat{x}_j^{(k)}$ in (13) indicates that, $y_j^{(k)}$ may be decoded into $\hat{x}_j^{(k)}$ with high confidence. In other words, $y_j^{(k)}$ is considered as the noisy version of $\hat{x}_j^{(k)}$. However, instead of instantly setting $y_j^{(k)}$ to be $\hat{x}_j^{(k)}$, cautious shift is operated on $y_j^{(k)}$ towards $\hat{x}_j^{(k)}$. While in the case of $\mathcal{M}(\hat{v}_j^{(k)}) \neq \hat{x}_j^{(k)}$, the received signal will be shifted towards the decision boundary of the two candidates, achieving a trade-off between the two choices. Here the decision boundary of $\mathcal{M}(\hat{v}_j^{(k)})$ and $\hat{x}_j^{(k)}$ is the bisector perpendicular to $\vec{\mathbf{L}}_j^{(k)} = \mathcal{M}(\hat{v}_j^{(k)}) - \hat{x}_j^{(k)}$.

Then using $\vec{\mathbf{L}}_j^{(k)}$ and $\xi_j^{(k)}$, the detector updates currently input vector according to (12). Based on $y_j^{(k+1)}$, new hard-decision is made, and the results are delivered to the variable nodes as the updated message passed from detector to the decoder. The whole working process of the IJDD algorithm is shown in Fig. 3.

In summary, the IJDD algorithm can be formulated as follows.

IJDD Algorithm

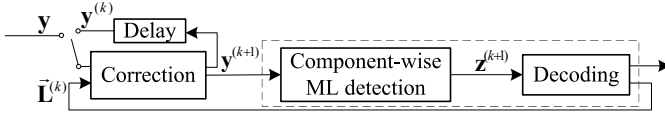


Fig. 3. Iterative joint detection-decoder.

- **Initialization.**

Set $k = 0$, and $\mathbf{y}^{(0)} = \mathbf{y}$.

- **Repeat the following while $k < k_{max}$**

- 1) **Signal detection:**

- For $j = 0, 1, \dots, N-1$, make ML decisions based on $y_j^{(k)}$, yielding $\hat{x}_j^{(k)}$ and $z_j^{(k)}$ as in (6) and (7).
- Pass the message $\mathbf{z}^{(k)}$ to LDPC decoder.

- 2) **Compute syndrome and do check-node update:**

- Compute syndrome vector $\mathbf{s}^{(k)} = \mathbf{z}^{(k)} \mathbf{H}^T$.
- If $\mathbf{s}^{(k)} = \mathbf{0}$, then terminate iteration and output $\hat{\mathbf{v}}^{(k)} = \mathbf{z}^{(k)}$ as the decoded codeword;
- else for $i = 0$ to $M-1$ and $j = 0$ to $N-1$, compute check-to-variable messages $\sigma_{i,j}^{(k)}$ as in (10).

- 3) **Variable-node update and correction:**

- For $j = 0$ to $N-1$,
- evaluate $(\hat{v}_j^{(k)}, \Delta f_j^{(k)}, f_j^{(k)}(a_{max}))$ based on $\{\sigma_{i,j}^{(k)}\}$ at variable nodes;
- send the message triples $(\hat{v}_j^{(k)}, \Delta f_j^{(k)}, f_j^{(k)}(a_{max}))$ to detector and perform signal corrections as done in Section III-B.

- 4) $k = k + 1$, **entering next iteration.**

- If $k = k_{max}$, declare a decoding failure.

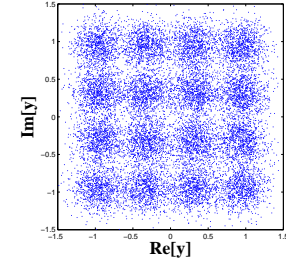
It is worth mentioning that in the IJDD algorithm differs from QSPA/FFT-QSPA in two aspects: 1) In the IJDD algorithm, only simple operations such as additions, comparisons, look-up tables, negligible amount of real operations and finite field operations are required; 2) The iterative process based on hard information is performed among detector/demapper, variable nodes and check nodes, while in the QSPA/FFT-QSPA, soft information propagates only between variable and check nodes.

From the above, it can be seen that the IJDD algorithm is easy to implement, and can achieve high speed. However, like existing reliability-based decoding algorithms, to ensure the reliability of majority voting, the column weights of \mathbf{H} have to be relatively large. It seems not easy for randomly constructed q -ary LDPC codes to fulfill this requirement, thus application of IJDD algorithm is restricted to q -ary LDPC codes constructed based on finite fields [10] or finite geometry [11].

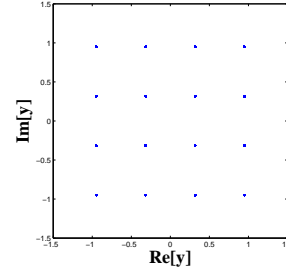
IV. SIMULATION RESULTS

In this section, two examples of q -ary LDPC codes are provided to demonstrate the effectiveness of the proposed IJDD algorithm.

Example 1: Consider a 16-ary (255, 175) regular LDPC code constructed based on finite fields. The factor graph of this



(a) Received signal space



(b) Corrected signal space

Fig. 4. Scatter plot of signal space before and after decoding with 10 iterations

code has 255 variable nodes and 255 check nodes (including 175 redundant check equations). Both the row and column weights are 16. With the use of 16-QAM signaling over the AWGN channel, scatter plots for signal vectors before and after decoding using IJDD with 10 iterations at $E_b/N_0 = 8.0$ dB are shown in Fig. 4(a) and Fig. 4(b), respectively. Clearly, the scatter plot in Fig. 4(b) corresponds to an ideal signal constellation, i.e., all received samples have been shifted to the probable originally transmitted constellation points.

Shown in Fig. 5 are the symbol and word error performances of this coded system decoded using the IJDD algorithm and FFT-QSPA with 50 iterations. It is seen that at a SER of 10^{-6} , the IJDD algorithm performs only 0.67dB away from the FFT-QSPA. Similar observation can be made for WER.

To illustrate the rate of decoding convergence of the IJDD algorithm, simulations were also carried out for $k_{max} = 10$ and $k_{max} = 5$, respectively, with results shown in Fig. 6. It is seen that with 10 and 50 iterations, the BER curves of the IJDD algorithm nearly overlap each other. Even with 5 iterations, the loss of performance is only 0.35 dB compared to IJDD with 50 iterations.

Example 2: Consider a 32-ary (1023, 781) regular LDPC code constructed based on finite fields. The row redundancy is 781 and both the row and column weights are 32. The performance of this code incorporated with 32-QAM modulation over the AWGN channel is illustrated in Fig. 7, where the codewords were obtained by encoding the randomly generated source data. Surprisingly, the IJDD algorithm outperforms the FFT-QSPA by 1.0 dB at a BER of 10^{-5} . This result may be caused by large column weights and large row redundancy of the code, which can offer high reliability for IJDD. In addition,

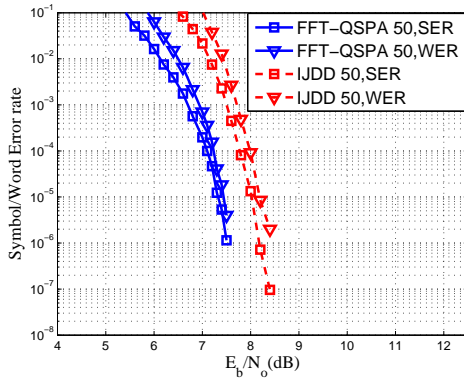


Fig. 5. Error performance of the 16-ary (255,175) LDPC code decoded with the FFT-QSPA and the IJDD algorithm (16-QAM)

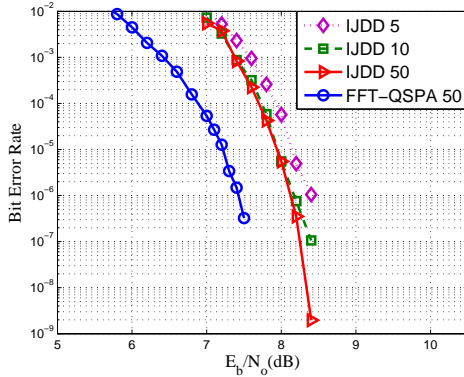


Fig. 6. Rate of decoding convergence of the IJDD algorithm for decoding the 16-ary (255,175) LDPC code (16-QAM)

the large column weights and large row redundancy of the code may make the FFT-QSPA algorithm not suitable for decoding it.

V. CONCLUSIONS

In this paper, we propose an iterative joint detection-decoding algorithm for q -ary LDPC-coded systems, which can be characterized as a hard-decision based message-passing algorithm and so has low computational complexity. For q -ary LDPC codes with large row redundancy and column weights, the proposed algorithm can offer good performance or even outperforms the FFT-QSPA with a lower computational complexity. Furthermore, the fast rate of decoding convergence of our proposed algorithm makes it particularly attractive, thus offering an attractive candidate for practical applications of q -ary LDPC codes. Although only regular LDPC codes are considered here, the proposed algorithm can be extended directly to decode irregular q -ary LDPC codes.

ACKNOWLEDGMENT

The authors would like to thank Prof. Shu Lin for his enlightening lectures. They also wish to thank Chao Chen for providing the parity-check matrices used in this paper, and Lin Zhou and Wei Lin for helpful discussions. This work is supported jointly by NSFC under Grants 60972046

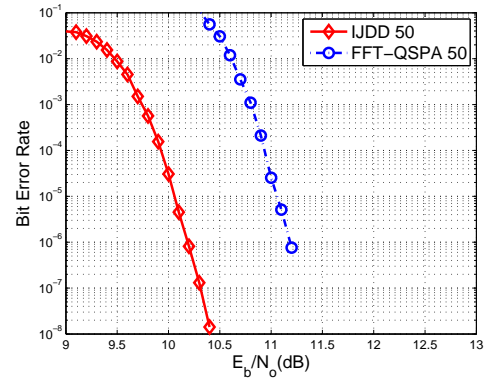


Fig. 7. Error performance of the 32-ary (1023,781) LDPC code decoded with the FFT-QSPA and the IJDD algorithm (32-QAM)

and U0635003, the National S&T Major Special Project (No. 2009ZX03003-011), and the PCSIRT (No. IRT0852).

REFERENCES

- [1] R. G. Gallager, *Low-Density Parity-check codes*. Cambridge, MA: MIT Press, 1963.
- [2] M. C. Davey and D. J. C. MacKay, "Low density parity check codes over $GF(q)$," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165-167, June 1998.
- [3] X.-Y. Hu and E. Eleftheriou, "Binary representation of cycle Tanner-graph $GF(2^q)$ codes," in *Proc. IEEE Int. Conf. Commun.*, Paris, France, Jun. 2004, pp. 528-532.
- [4] D. J. C. MacKay and M. C. Davey, "Evaluation of gallager codes for short block length and high rate application," in *Proc. IMA International Conference on Mathematic and its Applications: Codes, Systems and Graphical Models*, pp. 113-130, Springer-Verlag, New York, 2000.
- [5] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over $GF(q)$," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633-643, Apr. 2007.
- [6] S. Song, L. Zeng, S. Lin, and K. Abdel-Ghaffar, "Algebraic constructions of nonbinary quasi-cyclic LDPC codes," in *Proc. IEEE Int. Symp. Inform. Theory*, Seattle, WA, Jul. 2006, pp. 83-87.
- [7] V. Rathi, R. Urbanke, "Density evolution, thresholds and the stability condition for non-binary LDPC codes," *IEE Proc. Commun.*, vol. 152, no. 6, pp. 1069-1074, Dec. 2005.
- [8] A. Bannatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 549-583, Feb. 2006.
- [9] G. Li, I. J. Fair, and W. A. Krzymien, "Density Evolution for Nonbinary LDPC Codes Under Gaussian Approximation," *IEEE Trans. Inform. Theory*, vol. 55, no. 3, Mar. 2009.
- [10] L. Zeng, L. Lan, Y. Tai, S. Song, S. Lin, and K. Abdel-Ghaffar, "Constructions of nonbinary quasi-cyclic LDPC codes: a finite field approach," *IEEE Trans. Commun.*, vol. 56, no. 4, pp. 545-554, Apr. 2008.
- [11] L. Zeng, L. Lan, Y. Tai, S. Song, S. Lin, and K. Abdel-Ghaffar, "Constructions of nonbinary quasi-cyclic LDPC codes: a finite geometry approach," *IEEE Trans. Commun.*, vol. 56, no. 3, pp. 378-387, Mar. 2008.
- [12] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2004.
- [13] N. Mobini, A. H. Banihashemi and H. Hemati, "A differential binary message-passing LDPC decoder," *IEEE Trans. Commun.*, vol. 57, no. 9, pp. 2518-2523, Sep. 2009.
- [14] A. Nough and A. H. Banihashemi, "Bootstrap decoding of low-density parity-check codes," *IEEE Commun. Lett.*, vol. 6, no. 9, pp. 391-393, Sep. 2002.
- [15] Q. Huang, J. Y. Kang, L. Zhang, S. Lin and K. Abdel-Ghaffar, "Two reliability-based iterative majority-logic decoding algorithms for LDPC codes," *IEEE Trans. Commun.*, vol. 57, no. 12, Dec. 2009.
- [16] G. D. Forney Jr., "Codes on graphs: Normal realizations," *IEEE Trans. Inform. Theory*, vol. 47, pp. 520-548, Feb. 2001.